DTU Library

Description of the Risø Puff Diffusion Model

Mikkelsen, Torben

Publication date:
1982

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Mikkelsen, T. (1982). *Description of the Risø Puff Diffusion Model*. Risø National Laboratory. Risø-M No. 2361

RISØ-M-2361

DESCRIPTION OF THE RISØ PUFF DIFFUSION MODEL

Torben Mikkelsen

Abstract. The Risø National Laboratory, Roskilde, Denmark, atmospheric puff dispersion model is described. This three-dimensional model simulates the release of Gaussian pollutant puffs and predicts their concentration as they are diffused and advected downwind by a horizontally homogeneous, time-dependent wind. Atmospheric characteristics such as turbulence intensity, potential temperature gradient, buoyant heat flux and maximum mixing depth have been considered.

INIS descriptors: ADVECTION; AIR POLLUTION; BOUNDARY LAYERS; CLUSTER EMISSION MODEL; COMPUTERIZED SIMULATION; DIFFUSION; EARTH ATMOSPHERE; HEAT FLUX; METEOROLOGY; PLUMES; SPATIAL DISTRIBUTION; TEMPERATURE INVERSIONS; TRAJECTORIES; TURBULENCE; WIND

UDC 551.510.4 : 681.3.06

October 1982
Risø National Laboratory, DK 4000 Roskilde, Denmark

ISBN 87-550-0885-2
ISSN 0418-6435

Risø Repro 1982

TABLE OF CONTENTS

1. INTRODUCTION

The downwind transport and distribution of atmospheric pollutants from a isolated source over land or water has become an important environmental factor in today's society. The need to understand the distribution of smoke, unpleasant or potentially harmful foreign gases and perhaps radioactive debris from a nuclear power plant accident are becoming more and more essential for industrial operations and construction planning. The dispersion of such atmospheric pollutants is commonly modeled by a standard Guassian plume model which computes one-hour average plume characteristics.

The Meteorology Section of the Risø National Laboratory, Roskilde, Denmark, has recently developed a puff model for prediction and simulation of atmospheric pollutant diffusion.

The model considers individual puffs of pollutants with specific release rates that are advected by a horizontally homogeneous wind over a grid. The wind input may be either the measured wind from a single point, a spatial average or a wind simulation. The model simulates the instantaneous plume characteristics by adding a group of puffs, growing in size, as they advect with the wind. A Gaussian plume model, on the other hand, provides a time averaged concentration pattern based on a single time average wind vector. In the puff model, the plume advects with a time series of actual wind data. Thus, the puff model is able to predict time varying concentration distributions in actual changing wind conditions, making it an appropriate tool for dynamical computations of downwind dispersions of pollutants.

A basic comparison of a puff model simulation and a typical plume is illustrated in Fig. 1. Looking from above, the instantaneous behaviour of a plume being advected from a source by the wind is shown. The outer cone-shaped contours represent the outer limit of the plume boundary and are identical in both Figs. 1 (A) and (B).

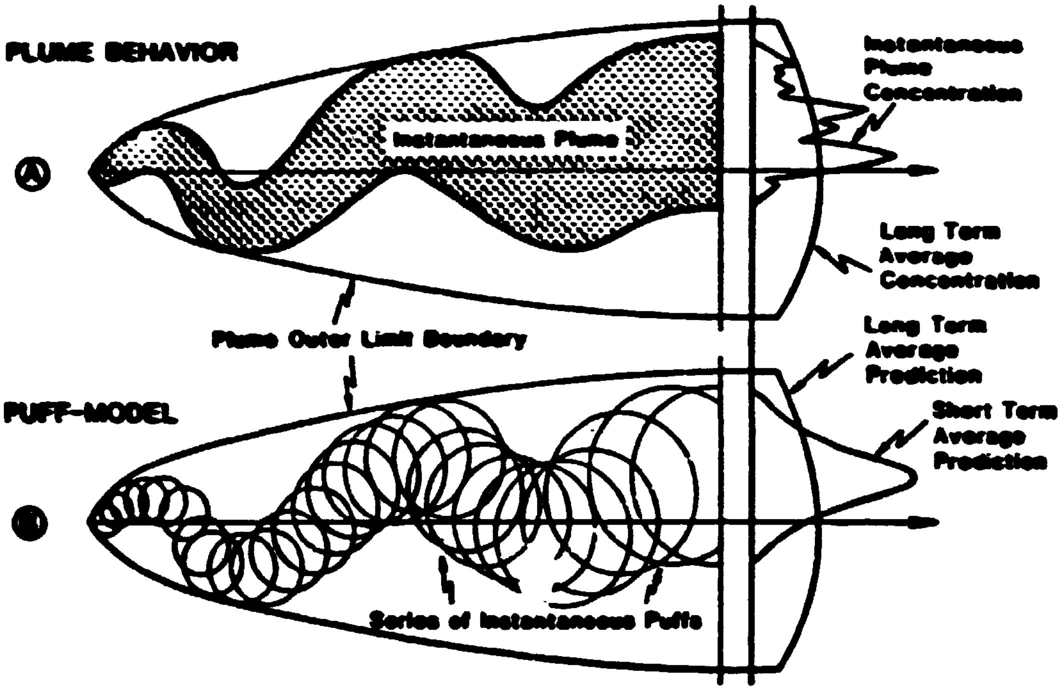


Fig. 1. Instantaneous behaviour of typical plume and a series of puffs from a puff model.

Fig. 1(A) shows an instantaneous depiction of an actual plume. The long-term average plume concentration is shown on the extreme right as a smooth curve with a maximum on the central axis. Also shown is the instantaneous plume concentration considered realistic but is of such a short time scale that it cannot be predicted or easily measured.

The puff model prediction is depicted in Fig. 1(B). The circles show the boundaries of individual puffs of pollutants released from the source. These puffs are advected and diffused downwind by a frequently updated wind. The long term average concentration prediction of the puff model is expected to be identical to the long term concentration of Fig. 1(A). The short term average pollutant prediction, a Gaussian curve shown on the extreme right, is not completely realistic but is a reasonable approximation to the instantaneous plume concentration profile.

2. RISØ PUFF MODEL

2.1 General characteristics

The Risø puff model is a three-dimensional computer model used for the prediction and/or simulation of the diffusion and advection of atmospheric pollutants. The puff model technique is to simulate a plume with Gaussian shaped puffs with specified release rates within a specified grid. The initial size of the puffs is normally one meter in diameter although this can be easily adjusted. The amount of material in a puff is the release rate times the elapsed time between puffs. Therefore, a long elapsed time between puff releases results in a higher initial puff pollutant concentration than a short time interval. This should not normally be of concern if an adequate balance is maintained between grid size, advection speed and puff release

The location of the puffs on the grid is determined by computing their movement for a finite time step using a measured wind field. The growth and buoyancy of the puffs are computed from simultaneous specifications of atmospheric turbulence intensity and stability and from buoyant heat flux at the source. An inversion cap through which pollutants cannot pass and the source height where pollutants are released are variable and can easily be adjusted. Grid distances within the model may vary from meters to kilometers and time durations from seconds to hours are possible.

This puff model has the capability of monitoring a maximum of twenty-five sources of puffs and its grid may contain up to 100 puffs. A puff source can be located anywhere on the grid and have a unique release rate, start and stop of release time, and heat production. When the center of a puff moves outside the boundaries of the grid (either horizontally or vertically), that particular puff is dropped from memory. In this way the model does not store irrelevant puff information, thus keeping computer memory requirements to a minimum.

A variable to control the amount of reflection/absorption of the pollutant by the surface is easily adjusted in the puff model. Such a paramter is of great value both in actual dispersion problems and also for gaining understandings of the plume/surface relationship.

The model calculates the concentration at each grid point by summing the contributions from surrounding puffs for each advection step. The grid concentrations can be allowed to accumulate or simply be updated with the latest instantaneous value. A minimum grid concentration of interest can be set to reduce computer run time by dropping concentrations too small to be of interest.

The output of the model contains periodic results of puff locations and concentrations as well as initial input verification. The time interval for the periodic results is adjusted by the input data. This recurrent lineprinter output contains:

- X-Y plane plots showing the position of the sources and of puffs inside the grid,
- X-Z plane plots of puff positions for evaluating plume rise for each vertical level of interest, and
- a table listing of the grid point concentrations for each level.

A computer drawn contour of the magnitudes of the pollutant concentrations is also available.

When considering distance between gridpoints (delta X,Y,Z), only spatial resolution and computer resources need be considered, calculated concentrations accuracy is not related to the grid-point separation. To ensure that no essential information on individual puffs is "hidden" between grid points, the grid separation should be adjusted dependent upon the size of the puffs at the downwind distance of interest. Other specific model configuration considerations are described in the following sections. They are also discussed in more detail in the model behaviour chapter.

2.2. Wind field

Once a puff is released, it is advected based upon wind data measurements at a single point only, normally the release point. This limits the validity of the model to situations where the wind field and turbulence can be assumed to be horizontally homogeneous throughout the grid. It is therefore important to ensure that the data obtained from such a single point measurement is representative of the wind structure for the whole area of interest.

The wind data are normally obtained in the form of a horizontal velocity time series. A vector sequence is formed by averaging over a convenient interval. These data are read into the model after being segregated into turbulence classes as discussed in the next section.

2.3 Turbulence intensity and diffusion

The growth/diffusion of a puff depends upon the turbulence intensity. To accoaunt for this growth, the puff model applies the theory for relative diffusion suggested by Smith and Hay (1961).

The turbulence intensity is defined to be the standard deviation of the wind direction (in radians) squared. These standard deviation values are collected for the same short time periods as the wind speed measurements used to advect the puffs. Therefore, the intensity of the turbulence which governs the relative diffusion of the puffs, can be adjusted along with the advecting wind speed after each time step, if the stability conditions changes.

A very low value of turbulence intensity represents a small standarad deviation, normally a stable atmosphere and a weak puff dispersion/diffusion. As the atmosphere becomes more unstable, the turbulence intensity increases along with an increase of theurind direction standard deviations and plume dispersion/diffusion. While these characteristics are resentative of turbulence over land, they can be applied to over water cases in a broad sense.

2.4 Plume rise

In the vertical direction, puff-rise can be accounted for by Briggs (1970) plume rise theory. In this case buoyancy is assumed to be conserved (adiabatic motion), and pressure forces, molecular viscosity and local density changes are considered small and are neglected. The rate at which a puff rises as it is advected downwind is a function of the buoyancy flux, wind speed, puff distance travelled and stability of the atmosphere. Plume rise is considered separately for each indvidual puff.

2.5 Reflection

The interaction of the pollutant with the surface is adjustable and can be easily changed in the input data. Total reflection or absorption or a fraction between the two can be used.

2.6 Limit of mixing depth

The effect of an atmospheric lid (inversion) can be applied in the model to limit the vertical movement of the pollutant. The model does not permit the plume to rise above this cap. When a mixing level is in effect, it acts to totally reflec the pollutant in the same manner as total reflection at the surface. This mixing cap also acts as an inversion limiting the vertical diffusion of a non-buoyant puff.

3. ACKNOWLEDGMENT

Stephan K. Rinard, The Naval Postgraduate School, Monterey, California is grateful acknowledged for, in connection with his master thesis: "An analysis of a puff dispersion model for a coastal region", to significantly improve the previous description of the Risø puff model (Mikkelsen, T. (1979) Simulation of obscuration smoke diffusion, 73 pp).

4. REFERENCES

Mikkelsen, T. (1979). Simulation of obscuration smoke diffusion. Work done under contract to the Danish Defence Research Establishment/Risø, 73 pp. Available from: Meteorology Section, Physics Department, Risø National Laboratory, DK-4000 Roskilde, Denmark.

Rinard, S.K. (1982). An analysis of a puff dispersion model for a coastal region. Master's Thesis from Naval Postgraduate School, Monterey, California, 88 pp.

APPENDIX A

Major Sections of the puff model

The Risø puff model code has previously been described Mikkelsen, T. (1979). The code also is well documented with comment statements. With that information and the outline to be provided in this and the following appendices, the computational and input/output procedures will be obvious.

The program and input data are stored on cards for the sake of permanency. For efficient operational execution, the program and input data cards are read on a disk within the computer. The model can then be run at will without reference to the original data cards. Minor changes can easeily be made directly on the disk both to the model and/or data before each execution.

The model can be separated into the following main sections:

a) Input data
b) Initial
c) Calculating
d) Output
e) Error diagnostics
f) Subroutines

These will be described separately in the following sections.

A. Input data section

The input data includes the variables shown in Table IV.

Table IV

Input Data Variables for the Puff Dispersion Model

Wind History	Potential Temperature Gradient
Turbulence Intensity	Buoyant Heat Flux
Grid Dimensions	Minimum Concentration of Interest
Mixing Depth	Reflection at Ground Level

Source Locations, Start/Stop Time, Strength, Heat Emission,
Number of Seconds between Advection Steps
Number of Secoands between Printouts/Plots
Number of Seconds between Puff Releases

The wind field and stability class for the current time step are read at the start of the calculation section.

The variables listed above are printed as input data check and a permanent record to accompany the actual output. In most cases the print command can be overridden by YES/NO options.

B. Initial section

Based upon the input data from section (A), the initial section specifies and initializes parameters to be used in the calculating section and is passed only once during execution of the model. The grid and some counters are initialized. Constants relating to reflectance, mixing depth and stability as well as those controlling the size of some of the loops within the model are established. Parameters such as number of puff releases per second, number of advection steps per second and number of advectionb steps per puff release are determined.

C. Calculation section

Using current wind and stability class data read at the start of the calculation section, the model advects the puff centers and calculates the growth rate and plume rise of the puffs. It removes the puffs that have left the grid (horizontally and/or vertically). The predicted concentration is computed at the grid points to include pollutants from all nearby puffs.

D. Output section

For time intervals designated by the input data, printer plots of the X-Y and Y-Z grid are produced. A maximum mixing level is marked on the Y-Z grid if in effect.

These plots include the source location and a trace of the plume from the release time t the maptime. Also printed at this interval is a X-Y table of grid concentrations for each vertical level of interest. These concentrations can be either accumulated or actual concentrations at the plot time.

Added to the puff model is a versatic plotter routine to smooth and contour the grid magnitude concentrations of the above tables.

E. Error Diagnostic section

If the model is directed by the input data beyond the limits of the design of the program, the program is terminated by way of the ereror diagnostic section. It prints comments relating to the commonly made input errors enabling the user to isolate problems.

F. Subroutines

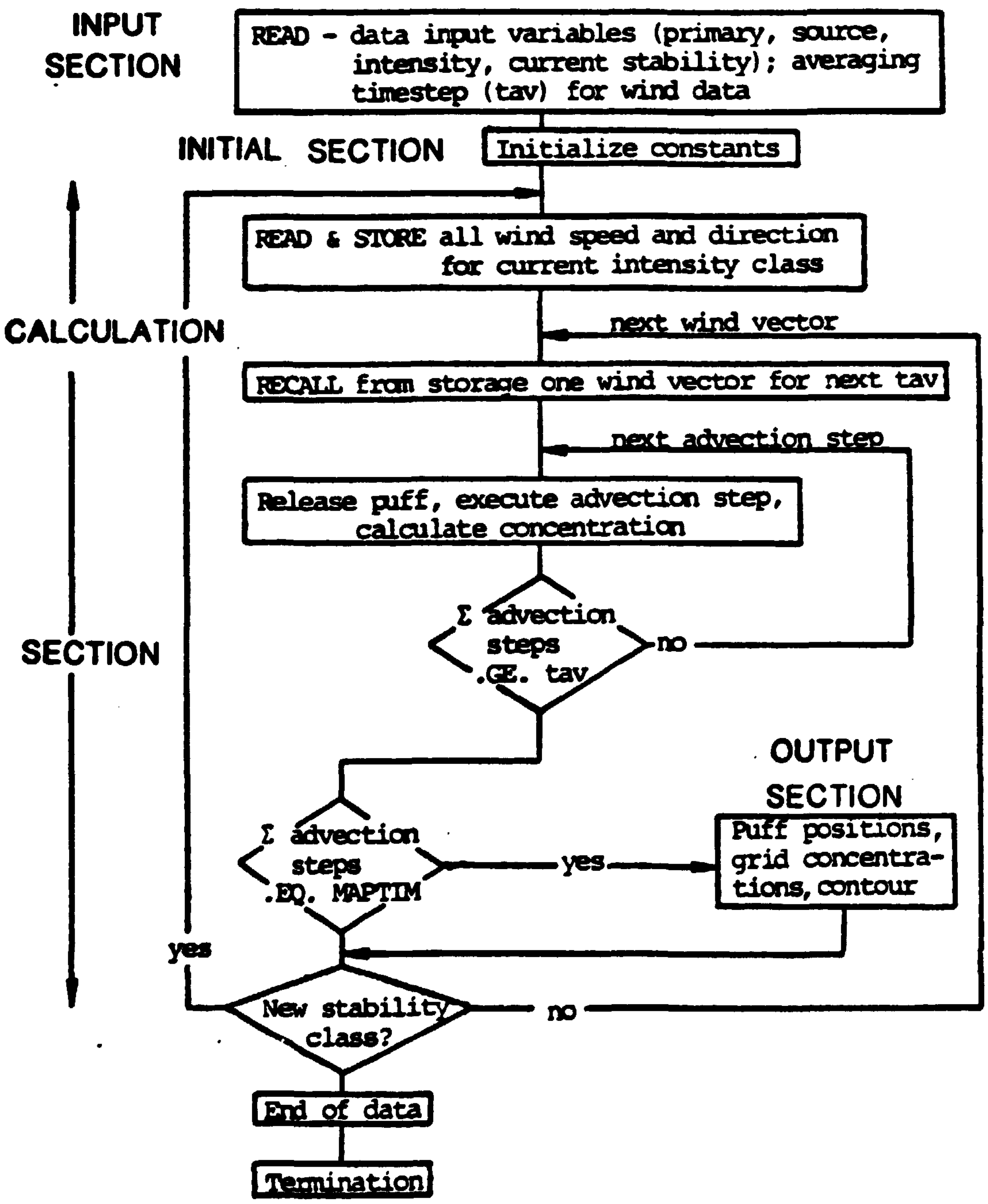
The subroutine "Sigris" calculates the puff size in the horizontal and vertical directions. It also estimates plume rise associated with pollutant buoyancy.

The subroutines "Ispace" and "Rspace" are used in the framework of the printer plots.

The subroutine "Plout" converts the plume concentrations to a logarithmic values, smoothes and then contours them using Risø inhouse contour subroutines. The values are converted to their logarithm values so that the problem of contouring over many orders of magnitude is simplified.2A

APPENDIX B

PUFF MODEL FLOW CHART



APPENDIX C

Puff Model Concentrations

CHEMIN -- Minimum grid concentration of interest
DELX, DELY, DELZ -- Distance in meters between grid points
DOSE -- Allows the concentration matrix to accumulate
DTDZ -- Potential temperature gradient (K/m) (.GE. 0)
HEAT -- Individual source heat emission (KWatt)
ICOLS -- Number of columns in grid (.LE. 10)
INST -- Instantaneous concentration matrix
ITIME -- Start time
JROWS -- Number of rows in grid
KPLANS -- Number of vertical levels in grid (includes surface)
MAPTIM -- Number of seconds between printer plots
NRELSE -- Number of seconds to stop of release
NRMULT -- Number of sources (.LE. 25)
NTADV -- Integer number of seconds between advection steps
REFLEC -- Reflection at ground level (0. none; 1.0 total)
SOURNR -- Number to identify source
SOURST -- Strength for individual source (g/s)
STOPRL -- Individual stource stop time (s)
STRTRL -- Individual source start time (s)
TAU -- Integer number of seconds between puff releases
TURN -- Angle of rotation of wind direction
XSOURCE -- X coordinate of source in grid units
YSOURCE -- Y coordinate of source in grid units
ZM -- Limited mixing depth (m)

APPENDIX D

LISTING OF PUFF MODEL COMPUTER CODE

```
// EXEC PGM=IEBGENER
//SYSPRINT DD DUMMY
//SYSIN DD DUMMY
//SYSUT2 DD UNIT=SYSDA,DISP=(NEW,PASS),
//   DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400),
//   SPACE=(TRK,(1,1)),DSN=&FT02
//SYSUT1 DD *
WIND DATA SEPT 29 81
810929 1630 #1800#
/D
" 085*04.7" 085*04.8
/
//  EXEC FRTXCLGN,NAME=CONRECQC
//FORT.SYSPRINT DD DUMMY
//FORT.SYSIN DD *
CFILE  1(KIND=DISK,TITLE='PRIMDA',FILETYPE = 7)                        PUF00010
CFILE  2(KIND=DISK,TITLE='VINDDA',FILETYPE=7)                          PUF00020
CFILE  3(KIND=DISK,TITLE='STABDA',FILETYPE=7)                          PUF00030
CFILE  4(KIND=DISK,TITLE='SOURCEDA',FILETYPE=7)                        PUF00040
CFILE  5(KIND=DISK,TITLE='INTSDA',FILETYPE=7)                          PUF00050
CFILE  PRINT(KIND=PRINTER,FILETYPE=7)                                  PUF00060
CFILE  6(KIND=PRINTER,FILETYPE=7)                                      PUF00070
C                                                                      PUF00160
C                                                                      PUF00170
C                                                                      PUF00180
C     THIS  MICRODIFFUSION  PROGRAM  REPRESENTS  THE STATE-OF-         PU00190
C     THE-ART CONCERNING THE DEVELOPMENT  OF A NUMERIC DIFFUSION       PU00200
C     MODEL  FOR OBSCURATION SMOKE.   ( RISO, MET. SEC. SEPT 1978)      UF00210
C                                                                      PUF00220
C     THE PROGRAM IS DOCUMENTED BY HEAVY USE OF COMMENT  STATEMENTS.   PUF00230
C                                                                      PUF00240
C     FOR COLLECTING AN OVERALL VIEW OF THE PROGRAM STRUCTURE, AS      PUF00250
C     WELL AS TO SET UP INPUT DATA FILES, IT IS ADVISED TO
C     CONSULT THE FLOWCHARTS AND DESCRIPTIONS IN THE CONSECUTIVE
C     REPORT.                                                          PUF00280
C                                                                      PUF00310
C                                                                      PUF00320
C                                                                      PUF00340
      COMMON HEATFX(25),I2,DMS,POINT,INTENS(14),STABPA,FBUFLX          PUF00350
     1,SPEED, CONST1,DELZ                                              PUF00360
      INTEGER TAU,POINT,ANGLE,XSB,XLB,YSB,YLB,ZSB,ZLB,YINT,XINTPF,YLINE PUF00370
     1,ZMG,ZINTPF,WINDAV,TOTTIM,SUMPUF,SOURNR,TPUFFS(25),XINT(100)     PUF00380
      INTEGER XSOURC(25),YSOURC(25), STRTRL(25),STOPRL(25)               UF00390
C
      LOGICAL LINE,COINCD,NOMXDP,GRRFLX                                PUF00410
      DIMENSION STRING(105),HORFRM(105),VERFRM(105),4RFRMZ(105),VRFRMZ(1PUF00420
     105),VERPLS(105),VRPLSZ(105),PARENT(105),NBUF(7),SBUF(7)          PUF00430
      REAL BL/' '/,SN1/'*'/,SN2/'I'/,SN3/'+'/,SN4/'-'/,SN5/'|'/,SN6/')'/PUF00440
```

```
      DIMENSION TITLE(18),WINDTX(18),PUFFTX(18),INTSTX(18),STABTX(18)   PUF00450
      DIMENSION PTABEL(25,100,7),SHIFT(100,7),CHI(50,50,10)             PUF00460

      DIMENSION ABC(10),SOURST(25),CPLOT(10,17),SOHT(25)                PUF00480

      REAL DATA(14),TYPE(14),INTENS                                     PUF00500
                                                                         UF00510
      DATA NO/'NO'/,BLANK/' '/,A/'A'/,B/'B'/,C/'C'/,D/'D'/,E/'E'/
      DATA PUNK/'.'/,ASTER/'*'/,ANFO/'"'/,SLASH/'/'/,DOSE/'DOSE'/       PUF00540
      DATA AA/'#'/                                                      PUF00550
C                                                                       PUF00560
C                                                                       PUF00570
C***********************************************************************PUF00580
C*****INPUT DATA SECTION************************************************PUF00590
C***********************************************************************PUF00600
C                                                                       PUF00610
C                                                                       PUF00630
C      INPUT DATA FROM DATA FILE PRIMDATA:                              PUF00640
C      PRIMARY DATA FOR PUFF MODEL                                      PUF00650
C                                                                       PUF00660
C      READ PRIMDA,CARD NO. 1:                                          PUF00670
C      READ(1,10)  ITIME,NRELSE,NSTEPS,ICOLS,JROWS,KP_ANS,NTADV         PUF00680
C\     READ PRIMDA,CARD NO. 2:                                          PUF00690
C      READ(1,20) MAPTIM,TAU,TURN                                       PUF00700
C      READ PRIMDA,TITLE STRING:                                        PUF00710
C      READ(1,30) TITLE                                                 PUF00720
C      READ PRIMDA,CARD NO. 4 - 5 :                                     PUF00730
C      READ(1,40) DELX,DELY,DELZ,CHEMIN,REFLEC                          PUF00740
C      READ PRIMDA,CARD NO. 5-9:                                        PUF00750
C      READ(1,50) ABC(5),ABC(6),ABC(7),ABC(8),ABC(9)                    PUF00760
C      END OF PRIMDA - INPUT FILE.                                      PUF00770
C                                                                       PUF00780

    1 FORMAT(2H0 )                                                      PUF00810
    2 FORMAT(2H1 )                                                      PUF00820
   10 FORMAT(7I5)                                                       PUF00830
   20 FORMAT(5I5)                                                       PUF00840
   30 FORMAT(18A4)                                                      PUF00850
   35 FORMAT(1H ,18A4)                                                  PUF00860
   40 FORMAT(3F10.2,E10.4,F10.5)                                        PUF00870
   50 FORMAT(A2/A2/A2/A2/A4)                                            PUF00880
  100 FORMAT(6X,31HKEY PARAMETERS FOR CURRENT RUN:)                     PUF00890
  200 FORMAT(6X,13HITIME  =     ,I5,6X,13HNRELSE =     ,I5,6X,13HNSTEPS PUF00900
     1=     ,I5)                                                        PUF00910
  300 FORMAT(6X,13HICOLS  =     ,I5,6X,13HJROWS  =     ,I5,6X,13HKPLANS PUF00920
     1=     ,I5)                                                        PUF00930
  400 FORMAT(6X,13HNTADV  =     ,I5,6X,13HMAPTIM =     ,I5,6X,13HTAU    PUF00940
     1=     ,I5)                                                        PUF00950
```

```
  600 FORMAT(6X,8HDELX    =,F10.2,6X,8HDELY    =,F10.2,6X,8HDELZ    =,F10.2
     1)
  700 FORMAT(6X,8HCHEMIN =,E10.4,6X,8HREFLEC =,F10.5,6X,13HTURN        =
     1    ,I5)
C     SKIPPING LINE PRINTING OF PRIMDATA IF SPECIFIED
      IF(ABC(5) .EQ.  NO ) GO TO 751
      DO  750 I = 1,5
  750 WRITE(6,1)
      WRITE(6,35) TITLE
      WRITE(6,1)
      WRITE(6,1)
      WRITE(6,100)
      WRITE(6,1)
      WRITE(6,200) ITIME,NRELSE,NSTEPS
      WRITE(6,1)
      WRITE(6,300) ICOLS,JROWS,KPLANS
      WRITE(6,1)
      WRITE(6,400) NTADV,MAPTIM,TAU
      WRITE(6,1)
      WRITE(6,600) DELX,DELY,DELZ
      WRITE(6,1)
      WRITE(6,700) CHEMIN,REFLEC,TURN
      WRITE(6,1)
C
  751 CONTINUE
C
C     READ SOURCEDATA INPUT FILE
C
  800 FORMAT(A1,I2,A1)
  810 FORMAT(5I5,3F10.5)
  820 FORMAT(48H CURRENT SOURCEDATA : NUMBER OF ACTIVE SOURCES :,I4)
C
C     READ PUFFDATA,TITLESTRING
      READ(4,30) PUFFTX
      WRITE(6,30) PUFFTX
C     READ NUMBER OF MULTISOURCES:NRMULT
      READ(4,800) ABC(3),NRMULT,ABC(4)
      WRITE(6,800) ABC(3),NRMULT,ABC(4)
C     INPUT FORMAT TESTING:
      IF(ABC(3).NE.AA.OR.ABC(4).NE.AA) GO TO 8920
C     READ INDIVIDUAL SOURCEDATA:
C
C     SETTING FRAMEDATA: (M) SMALL-X, (M) FULL-X, ETC.
      MFX = ICOLS
      MSX = 1
      MFY = JROWS
      MSY = 1
      MFZ = KPLANS
```

```
      MSZ = 1
      XSB=MSX-1
      XLB=MFX-1
      YSB=MSY-1
C     BOTTOM OF FRAME : YBB = YSB - .9
      YBB = YSB - .9
      ZSB = MSZ - 1
      ZLB = MFZ - 1
      YLB=MFY-1
      YLL=YLB + 0.1
      DO 850 I = 1,NRMULT
      READ(4,810)  SOURNR,XSOURC(I),YSOURC(I),STRTRL(I),
     2STOPRL(I),SOURST(I),HEATFX(I),SOHT(I)
      WRITE(6,810) SOURNR,XSOURC(I),YSOURC(I),STRTRL(I),
     2STOPRL(I),SOURST(I),HEATFX(I),SOHT(I)
C     TESTING INDIVIDUAL SOURCEDATA:
      IF(I.NE.SOURNR) GO TO 8910
C     OFF GRID TEST FOR SOURCE COORDINATES:
      IF(XSOURC(I).GT.XLB.OR.XSOURC(I).LT.XSB.OR.YSDJRC(I).GT.YLB.OR.YSO
     1URC(I).LT.YSB) GO TO 9000
  850 CONTINUE
C
      WRITE(6,2)
      WRITE(6,1)
      WRITE(6,35) PUFFTX
      WRITE(6,1)
      WRITE(6,820) NRMULT
      WRITE(6,1)
C
C     SETTING UP STRING VARIABLES FOR PLOTTING PURPOSES
      DO 911 N = 1,105
      STRING(N)=BL
      VERFRM(N)=BL
      VERPLS(N) = BL
  911 HORFRM(N) = BL
      NY1=MFX*10
      DO 916 I = MSX,NY1
      NY2=I+4
      DO 915 NN = I,NY2
  915 HORFRM(NN)= SN4
      HORFRM(I+5) = SN2
      NY3=I+6
      NY4=I+9
      DO 916 NN = NY3,NY4
  916 HORFRM(NN)= SN4
      VERFRM(10*MFX  + 3) = SN5
      VERPLS(10*MFX  + 3) = SN3
C
```

```
C      OUTPRINTING CURRENT SOURCE POSITION(S) IN GRID PICTURE             PUF01920
C                                                                         PUF01930
C      SKIP PLOT OF SOURCE POSITIONS IF SPECIFIED IN PRIMDA               PUF01940
       IF(ABC(6) .EQ.  NO ) GO TO 999                                     PUF01950
C                                                                         PUF01970
       IF(ICOLS.GT. 10) GO TO 995                                         PUF01980
  860  FORMAT(1H ,49X,33H CURRENT SOURCE DATA AS SPECIFIED,/50X,27H IN SOPUF01990
      1URCE DATA INPUT FILE:,/)                                           PUF02000
  865  FORMAT(1H0,50X,'SOURCES ARE REPRESENTED BY:',/55X,'SOURCE NUMBER', UF02010
      1/55X,'START TIME (SEC)',/55X,'STOP TIME (SEC)',/55X,
      2,'SOURCE STRENGTH',/55X,'BUOYANT HEAT FLUX.'//)

  870  FORMAT(2H0 ,16H Y COORDINATE OF,25X,32H X COORDINATE OF THE GRID PPUF02050
      1OINTS/2X,16H THE GRID POINTS,I8,9I10/)                             PUF02060
  871  FORMAT(2H0 ,16H Y COORDINATE OF,25X,32H Z COORDINATE OF THE GRID PPUF02070
      1OINTS/2X,16H THE GRID POINTS,I8,9I10/)                             PUF02080
C                                                                         PUF02090
       WRITE(6,860)                                                       PUF02100
       WRITE(6,865)                                                       PUF02110
       WRITE(6,870)   (I,I=XSB,XLB)                                       PUF02120
C      WRITING DATA INTO GRIDPOINTS:                                      PUF02130
  910  FORMAT(1H+,I11,5X,2H +,105A1)                                      PUF02140
  912  FORMAT(1H+, 19X, 105A1)                                            PUF02150
  913  FORMAT(1H ,17X,1H|,105A1/1H ,17X,1H|,105A1)                        PUF02160
  914  FORMAT(1H+ ,17X,1H|,105A1)                                         PUF02170
C                                                                         PUF02190
C                                                                         PUF02200
       WRITE(6,1)                                                         PUF02210
       WRITE(6,912) HORFRM                                                PUF02220
       WRITE(6,913) VERFRM,VERFRM                                         PUF02230
       MAX =JROWS -1                                                      PUF02240
       NY5=MAX+1                                                          PUF02250
       DO 950     NY6=1,NY5                                               PUF02260
       I=NY6-1                                                            PUF02270
       MAXMI   = MAX - I                                                  PUF02280
       WRITE(6,910)  MAXMI, VERPLS                                        PUF02290
       DO 920 J=1,NRMULT                                                  PUF02300
       IF(MAX-I .NE. YSOURC(J) )  GO TO 920                               PUF02310
       CALL ISPACE(XSOURC(J),J)                                           PUF02320
  920  CONTINUE                                                           PUF02330
       WRITE(6,913) VERFRM,VERFRM                                         PUF02340
C                                                                         PUF02350
       DO 932 J = 1,NRMULT                                                PUF02360
       IF(MAX-I .NE.YSOURC(J)) GO TO 932                                  PUF02370
       WRITE(6,914) VERFRM                                                PUF02380
       CALL ISPACE(XSOURC(J),STRTRL(J))                                   PUF02390
  932  CONTINUE                                                           PUF02400
       WRITE(6,913) VERFRM,VERFRM                                         PUF02410
```

```
C
      DO 934 J = 1,NRMULT
      IF(MAX-I .NE. YSOURC(J)) GO TO 934
      WRITE(6,914) VERFRM
      CALL ISPACE(XSOURC(J),STOPRL(J))
 934  CONTINUE
      WRITE(6,913) VERFRM,VERFRM
C
      DO 940 J=1,NRMULT
      IF(MAX-I .NE. YSOURC(J) )  GO TO 940
      WRITE(6,914) VERFRM
      CALL RSPACE(XSOURC(J),SOURST(J))
  940 CONTINUE
      WRITE(6,913) VERFRM,VERFRM
C
      DO 930 J=1,NRMULT
      IF(MAX-I .NE. YSOURC(J) )  GO TO 930
      WRITE(6,914) VERFRM
      CALL RSPACE(XSOURC(J),HEATFX(J))
  930 CONTINUE
      WRITE(6,913) VERFRM,VERFRM
      WRITE(6,913) VERFRM,VERFRM
C
  950 CONTINUE
      WRITE(6,1)
      WRITE(6,912) HORFRM
C
      GO TO 999
  990 FORMAT(53H SOURCE DATA PLOT SUPPRESSED BECAUSE"ICOLS"EXCEEDS 10)
  995 WRITE(6,990)
  999 CONTINUE
C
C     DEFINE STABILITY AND INTENSITY CLASSES
C     INPUT FROM INTENSITY - DATA: INTSDA
C
  960 FORMAT(14 F5.4)
  965 FORMAT(1H0, 46H IN THE CURRENT RUN, THE STABILITY-CLASSES ARE,/41H
     1  CONNECTED TO INTENSITY DATA AS FOLLOWS:)
  970 FORMAT(1H ,21H STABILITY CLASS NO.:,I3,13I5)
  975 FORMAT(1H ,21H INTENSITY DATA     :, 14F5.4)
C
C     READ INTSDA,TITLE-STRING:
      READ(5,30) INTSTX
C     WRITE(6,30) INSTX
C     READ INTSDA, NO. OF INTENSITY-CLASSES: NRINCL
      READ(5,800) ABC(3),NRINCL,ABC(4)
C     WRITE(6,802) NRINCL
C     INPUT FORMAT TESTING:
```

```
  802 FORMAT(1X,I5)
      IF(ABC(3).NE.AA.OR.ABC(4).NE.AA) GO TO 8890
C     READ INTENSITY-CLASSES INTO REAL ARRAY: INTENS
      READ(5,960,END=801) (INTENS(I),I=1,NRINCL)
C     WRITE(6,960) (INTENS(I),I=1,NRINCL)
C
C     OUTPRINTING CURRENT INTENSITY CLASSES:
C
C     SKIP PRINTING OF INTENSITY DATA IF SPECIFIED IN PRIMDA
  801 IF(ABC(7).EQ. NO ) GO TO 980
      WRITE(6,2)
      WRITE(6,35) INTSTX
      WRITE(6,1)
      WRITE(6,965)
      WRITE(6,970) (I,I=1,NRINCL)
      WRITE(6,975) (INTENS(I),I=1,NRINCL)
      WRITE(6,1)
      WRITE(6,1)
  980 CONTINUE
C
C     END OF INTENSITY DATA SECTION.
C
C     INPUT FROM STABILITY DATA:STABDA
C
C     READ STABDA,TITLESTRING:
      READ(3,30) STABTX
C     READ STABDA,POTENTIAL TEMPERATURE GRADIENT (>0).
      READ(3,889) DTDZ
C     READ STABDA,LIMIT OF MIXING DEPTH: ZM (METERS).
      READ(3,992) ZM
C     INDATA-TEST ON ZM:
      IF(AMOD(ZM,DELZ) .NE. 0.)GO TO 8880
C
C     OUTPRINTING CURRENT STABILITY-DATA:
  889 FORMAT(F10.4)
  991 FORMAT(1H0,45H IN THE CURRENT RUN,THE POTENTIAL TEMPERATURE/21H  G
     1RADIENT IS SET TO:,F10.4)
  992 FORMAT(F10.2)
  993 FORMAT(1H0,36H NO FINAL MIXING DEPTH IS SPECIFIED.)
  994 FORMAT(1H0,32H THE MIXING LAYER IS LIMITED AT:,F10.2,8H METERS.)
      WRITE(6,1)
      WRITE(6,1)
      WRITE(6,1)
      WRITE(6,35) STABTX
      WRITE(6,1)
      WRITE(6,1)
      WRITE(6,991) DTDZ
      WRITE(6,1)
```

```
      WRITE(6,1)
      IF(ZM .EQ. 0.) WRITE(6,993)
      IF(ZM .GT. 0.) WRITE(6,994) ZM
C
C     END OF STABILITY DATA SECTION.
C
C
C     INPUT FROM WINDDATA:
 1110 FORMAT(I6,1X,I5,1X,1A1,I4,1A1)
 1120 FORMAT(1H0,20HCURRENT WINDDATA: STARTDATE=,I6,4X,10HSTARTHOUR=,I4,
     14X,14HWINDAV.(SEC.)=,I4)
C     READ WINDDA,TITLE STRING:
      READ(2,30) WINDTX
C     READ WINDDA,STARTTIME AND WINDFIELD AVERAGING TIME:
      READ(2,1110) DATE,STRTHR,ABC(1),WINDAV,ABC(2)
C     INPUT FORMAT TESTING:
      IF(ABC(1).NE.AA.OR.ABC(2).NE.AA) GO TO 8990
      WRITE(6,2)
      WRITE(6,1)
      WRITE(6,1)
      WRITE(6,1)
      WRITE(6,35) WINDTX
      WRITE(6,1)
      WRITE(6,1120) DATE,STRTHR,WINDAV
C
C     TESTING ON SPECIFIED TIME INCREMENTS: TAU,NTADV,WINDAV:
      ITEST1 = MOD(TAU,NTADV)
      IF(ITEST1.NE.0) GO TO 8980
      ITEST2 = MOD(WINDAV,NTADV)
      IF(ITEST2.NE.0) GO TO 8970
C     END OF FIXED WINDDATA SPECIFICATIONS.
C
C
C
C*****************************************************
C*****INITIAL SECTION*********************************
C*****************************************************
C
C     (THIS PART OF THE PROGRM IS ONLY PASSED ONCE.)
C
 1130 FORMAT( 14(A1,A4))
C     COUNTER FOR STABILITY SPECIFICATIONS GIVEN BY WINDDATA: NRSTAB
      NRSTAB =0
C     INITIATING A THREE DIMENSIONAL GRID: CHI
      DO 1200 I = 1,ICOLS
      DO 1200 J = 1,JROWS
      DO 1200 K = 1,KPLANS
 1200 CHI(I,J,K) = 0
C
C
```

```
C      NUMBER OF PUFF RELEASES PER SEC: TAUINVERS.
       TAUINV = 1.0/ FLOAT(TAU)
C
C      NUMBER OF ADVECTION STEPS PER SEC.: ADSTPS.
       ADSTPS = 1.0/ FLOAT(NTADV)
C      BASIC DOSE PER PUFF:(GRAM/SEC.)*TAU  =  GRAM/PJFF.
       BADOPP = 1*TAU
C
C      NUMBER OF BASIC ADVECTION STEPS (INTEGER NUMBER) PER PUFF RELEASE:
       NADPRP = TAU/NTADV
C
C      NUMBER OF BASIC ADVECTION STEPS (INTEGER NUMBER) PER WINDFIELDSP.
       NADPRW= WINDAV/NTADV
C
C

C      TOTAL RUNTIME COUNTER: TOTTIM.
       TOTTIM =0
C
C
C      COUNTER FOR REMOVED PUFFS: LEAVE
       LEAVE = 0
C      STABILITY PARAMETER FOR PLUMERISE:
C      STABPA = G/T*(DTHETE/DZ)
       STABPA = .033*DTDZ
C      CONSTANT IN CONNECTION WITH PLUMERISE FORMULA FOR USE IN
C      SUBROUTINE SIGRIS: CONST1.
       CONST1 = 0.6667 * 1.6**1.5
C
C      IF MIXING DEPTH IS NOT SPECIFIED,SET NOMXDP = .TRUE.
       IF(ZM .EQ. 0.) NOMXDP = .TRUE.
C
C      IF REFLECTANCE AT GROUND LEVEL IS SPECIFIED,SET GRRFLX = .TRUE.
       IF(REFLEC .GT. 0.) GRRFLX = .TRUE.
C      MIXING DEPTH IN GRID-UNITS: ZMG
       ZMG = ZM/DELZ
C      TESTING THAT MIXING DEPTH IS INSIDE GRID:
       IF(ZMG .GT. (KPLANS -1)) GO TO 8870
C
C      END OF INITIAL SECTION
C
C
C***********************************************
C*****CALCULATION SECTION**********************
C***********************************************
C
C      READING STABILITY CLASS AND WINDDATA FROM INPUTFILE:
 1135  READ (2,1130)  (  TYPE(I),DATA(I)  , I = 1,14)
```

```
      BACKSPACE 2
      IF(TYPE(1).EQ.ANFO) READ(2,1131)(NBUF(I),SBUF(I),I=1,7)
      IF(TYPE(1).EQ.ANFO)WRITE(6,1131)(NBUF(I),SBUF(I),I=1,7)
      IF(TYPE(1).EQ.SLASH) READ(2,1130)
 1131 FORMAT(7(1X,I4,1X,F4.1))
C     LOOP THRU WINDDATA AT SPECIFIED TIMESTEPS
      I = 1
      IF(TYPE(1).NE.SLASH)GO TO 1150
      NRSTAB = NRSTAB + 1
C
C     COUNTING NUMBER OF WINDDATA SPECIFICATIONS: IWDASP
      IWDASP = 0
C     READING STABILITY CATEGORY FROM WINDDATA:
      CLASS =DATA(1)
      IF(CLASS.EQ. A )  POINT =1
      IF(CLASS.EQ. B )  POINT =2
      IF(CLASS.EQ. C )  POINT =3
      IF(CLASS.EQ. D )  POINT =4
      IF(CLASS.EQ. E )  POINT =5
      IF(CLASS.EQ.PUNK)GO TO 8930
      IF(CLASS.EQ.BLANK) GO TO 8940
 1140 FORMAT(53H PROGRAM STOPPED ORDINARILY FM WINDDATA SPECIFICATION)
      WRITE(6,1)
      WRITE(6,1141) NRSTAB,POINT
      WRITE(6,1)
 1141 FORMAT(4H THE,I3,38H. STABILITY SPECIFICATION CLASS IS NO.,I1)
      GO TO 1135
C
C     INPUT STRUCTURE TEST:
 1150 IF(TYPE(I).NE.ANFO .OR.TYPE(I+1).NE.ASTER) GO TO 1160
C
      IWDASP  = IWDASP + 1
C     CURRENT WINDDATA:
      JI=(I+1)/2
      ANGLE=NBUF(JI)
      SPEED=SBUF(JI)
      GO TO 1175
 1160 IF(TYPE(I).NE.BLANK.OR.TYPE(I+1).NE.BLANK) GO TO 8950
C     READ NEW DATA IN LINE 1135
      GO TO 1135
C     INDATA PART OF PROGRAM TERMINATED.
 1175 CONTINUE
C
C     CURRENT WINDDATA PRESENT.
C
C     OUTPRINTING CURRENT WINDDATA:
      WRITE(6,1161)  IWDASP ,ANGLE, SPEED
C
```

```
C                                                                       PUF04790
C     CALCULTING WIND VELOCITY IN GRID UNITS: VGX,VGY                   PUF04800
      VGX = SPEED*(COS(ANGLE*3.142/180)) / DELX                         PUF04810
      VGY = SPEED*(SIN(ANGLE*3.142/180)) / DELY                         PUF04820
C                                                                       PUF04830
C     RENAMING WIND AVERAGING TIME:WINDAV AS TAV:                       PUF04840
      TAV = WINDAV                                                      PUF04850
 1161 FORMAT(4H THE,I4,49H WINDDATASET IN THE CURRENT STAB.CLASS IS: ANGPUF04860
     1LE=,I4,8H ,SPEED=,F4.1)                                           PUF04870
C                                                                       PUF04880
C     LOOP THRU BASIC ADVECTION STEPS WITH CURRENT WIND FIELD           PUF04900
C
      DO 5000   NN=1,NADPRW                                             PUF04920
C
C     JUMPING OVER "ZERO-SETTING" OF CONCENTRATION MATRIX : CHI , IF    PUF04940
C     "DOSE MODE" IS SPECIFIED IN PRIMDA.                               PUF04950
      IF(ABC(8).EQ. DOSE ) GO TO 1256                                   PUF04960
C                                                                       PUF04970
      DO 1255 IG=1,ICOLS                                                PUF04980
      DO 1255 JG=1,JROWS                                                PUF04990
      DO 1255 KG=1,KPLANS                                               PUF05000
 1255 CHI(IG,JG,KG) = 0.0                                               PUF05010
C                                                                       PUF05020
 1256 CONTINUE                                                          PUF05030
C                                                                       PUF05040
C     TIMECOUNTER:TOTTIM (SEC.)                                         PUF05050
      TOTTIM = TOTTIM + NTADV                                           PUF05060
C     SKIPPING RELEASE-SECTION IF SPECIFIED
      IF(TOTTIM .GT. NRELSE) GO TO 1250                                 PUF05080
C     TESTING IF RELEASE CONDITIONS ARE FULFILLED                       PUF05090
      IF(MOD(TOTTIM,TAU) .NE. 0) GO TO 1250                             PUF05100
C                                                                       PUF05110
C     LOOP THRU MULTIPLE SOURCES                                        PUF05130
      DO 1250 I2 = 1,NRMULT                                             PUF05140
C                                                                       PUF05150
C     INDIVIDUAL RELEASE CONTROL AS SPECIFIED IN SOURCE DATA:           PUF05170
C                                                                       PUF05180
      IF((TOTTIM.LT.STRTRL(I2)) .OR. (TOTTIM.GT.STOPRL(I2))) GO TO 1250 PUF05190
C     TOTAL NUMBER RELEASED FROM SOURCE(I2): TPUFFS(I2):                PUF05200
      TPUFFS(I2) = TPUFFS(I2) + 1                                       PUF05210
C                                                                       PUF05220
C     SHIFTING PUFF TABLE ONE POSITION TO THE RIGHT AND THEREBY         PUF05230
C     GIVING SPACE FOR ONE NEW PUFF:                                    PUF05240
C                                                                       PUF05250
      J=1                                                               PUF05260
 1204 DO 1205 K=1,7                                                     PUF05270
 1205 SHIFT(J+1,K) = PTABEL(I2,J,K)                                     PUF05280
      J = J + 1                                                         PUF05290
```

```
      IF(J.GE.100) GO TO 8900
      IF(PTABEL(I2,J,1) .NE. 0) GO TO 1204
      DO 1210  L = 2,J
 1209 DO 1210 K = 1,7
 1210 PTABEL(I2,L,K) = SHIFT(L,K)
C
C     INSERTING NEW PUFF DATA IN PUFF TABLE AT J = 1
      PTABEL(I2,1,1) = TPUFFS(I2)
C     DOSE RELEASED WITH EACH PUFF: SPECIFIED SOURCE STRENGTH*SEC.
C     BETWEEN RELEASES
      PTABEL(I2,1,2) = SOURST(I2) * TAU
C
C     LOADING IN INITIAL SOURCE POSITIONS
      PTABEL(I2,1,3) = XSOURC(I2)
      PTABEL(I2,1,4) = YSOURC(I2)
C
C   TO AVOID NUMERICAL PROBLEMS IN ESTIMATING PLUME RISE,
C     SET SOHT(I2) (SOURCE HEIGHT) .GE. 1 METER.
      PTABEL(I2,1,5) =SOHT(I2)/DELZ
C     INITIAL SIZE OF PUFFS:
C     SIGMAXY SET TO 1 METER:
      PTABEL(I2,1,6) = 1
C     SIGMAZ SET TO 1 METER:
      PTABEL(I2,1,7) =1
C     END OF PUFF RELEASE SECTION.
 1250 CONTINUE
C
C     ADVECTION OF ALL PUFF CENTERS
C
C     ADVANCE OF PUFF CENTERS IN GRID UNITS (HORIZONTALLY)
      DGX = VGX* NTADV
      DGY = VGY* NTADV
C     TOTALLY TRAVELED  DISTANCE BY THE PUFFS IN METERS
C     DURING CURRENT BASIC ADVECTION STEP: DMS
      DMS =  SQRT((DGX*DELX)**2  +  (DGY*DELY)**2)
C
C     ADVECTION SECTION FOR ALL EXISTING PUFFS:
C     LOOP THRU ALL SOURCES,COUNTING REMOVED PUFFS:LEAVE
      DO 1300     I2 = 1, NRMULT
      J = 1
C     SKIPPING SOURCE I2,IF THE LAST BORN PUFF HAS LEFT GRID
      IF(PTABEL(I2,1,1).EQ.0) GO TO 1300
 1260 PTBL3  = PTABEL(I2,J,3) + DGX
      PTBL4  = PTABEL(I2,J,4) + DGY
C
C
C     CALLING SUBROUTINE "SIGRIS" ,THEREBY ADDING DEVIATION INCREMENT
C     AND PLUME RISE INCREMENT TO PUFF TABLE:
```

```
C       PTABEL(I2,J,5): Z-POSITION IN GRIDUNITS                          PUF05870
C       PTABEL(I2,J,6): SIGMAXY IN METERS                                PUF05880
C       PTABEL(I2,J,7): SIGMAZ IN METERS                                 PUF05890
C                                                                        PUF05900
        CALL SIGRIS(PTABEL(I2,J,5),PTABEL(I2,J,6),PTABEL(I2,J,7))        PUF05910
C                                                                        PUF05920
C       INTRODUCING AN UPPER LIMIT FOR BUOYANCY CONVECTION: ZM.          PUF05940
        IF(.NOT.NOMXDP.AND.PTABEL(I2,J,5).GT.ZMG) PTABEL(I2,J,5) = ZMG   PUF05950
C                                                                        PUF05980
C       Z - POSITIONS IN GRIDUNITS: PTBL5                                PUF05990
        PTBL5  = PTABEL(I2,J,5)                                          PUF06000
C                                                                        PUF06010
C       TESTING AND REMOVING PUFFS WHICH HAVE LEFT THE GRID:             PUF06020
        IF(PTBL3.GT.XSB.AND.PTBL3.LT.XLB.AND.PTBL4.GT.YBB.AND.PTBL4.LE.YL PUF06030
       1L.AND.PTBL5.LT.ZLB) GO TO 1290                                   PUF06040
C                                                                        PUF06050
C       REMOVE SECTION                                                   PUF06060
        LEAVE = LEAVE + 1                                                PUF06070
        IF(PTABEL(I2,J+1,1) .EQ. 0) GO TO 1265                           PUF06080
C       REMOVING PUFF BORN AT SOURCE I2 WHICH IS NOT THE LONGEST LIVING: PUF06090
C       LEFT JUSTIFICATION OF OLDER PUFFS:                               PUF06100
        JJ = J + 1                                                       PUF06110
 1269   DO 1270  K = 1,7                                                 PUF06120
 1270   SHIFT(JJ,K) = PTABEL(I2,JJ,K)                                    PUF06130
        JJ = JJ + 1                                                      PUF06140
        IF(PTABEL(I2,JJ,1) .NE. 0)  GO TO 1269                           PUF06150
        SHIFT(JJ,1) = 0                                                  PUF06160
        JMAX = JJ                                                        PUF06170
C       COPY SHIFT BACK INTO PTABEL:                                     PUF06180
        NY7=JMAX-1
        DO 1275 JJ = J,NY7
        DO 1275 K=1,7                                                    PUF06200
 1275   PTABEL(I2,JJ,K) = SHIFT(JJ+1,K)                                  PUF06210
C                                                                        PUF06220
C       RETURNING TO INCREMENTAL PART WITHOUT INCREASE IN J:             PUF06230
        GO TO 1260                                                       PUF06240
C                                                                        PUF06250
C       REMOVING LONGEST LIVING PUFF FROM SOURCE(I2):                    PUF06260
 1265   PTABEL(I2,J,1) = 0                                               PUF06270
C       CONTINUING WITH NEXT SOURCE                                      PUF06280
        GO TO 1300                                                       PUF06290
C                                                                        PUF06300
C       REPLACING NEW PUFF POSITION IN PUFF TABLE                        PUF06330
 1290   PTABEL(I2,J,3) = PTBL3                                           PUF06340
        PTABEL(I2,J,4) = PTBL4                                           PUF06350
C                                                                        PUF06380
C       CALCULATING GRID CONCENTRATION IN EACH BASIC ADVECTION STEP      PUF06390
C                                                                        PUF06420
```

```
C     RENAMING ESSENTIAL PARAMETERS:
C     DOSE IN CURRENT PUFF:
      QI = PTABEL(I2,J,2)
C     SIGMA VALUES IN METERS:
      SIGMXY = PTABEL(I2,J,6)
      SIGMZ  = PTABEL(I2,J,7)
C
C     CALCULATING MAXIMUM CONCENTRATION IN EACH PUFF CENTER
C     (PUFF-CHI-CENTER) , IN DIMENSION: GRAM/M**3 :
C     CONSTANT : (2*PHI)**(3/2)
      CONST  = 15.7496
C
      PCHCEN =  QI/(CONST*SIGMZ*SIGMXY**2)
C
C     SKIPPING SUMMATION SECTION IF CONCENTRATION IS TOO LOW
      IF(PCHCEN.LT.CHEMIN) GO TO 1500
C
C     CALCULATING MAXIMUM RADIUS OF INTEREST FOR EAC4 PUFF:
C     MAXIMUM PUFF RADIUS IN METERS:
      PFRMXY = SIGMXY * SQRT(-2.*ALOG(CHEMIN/PCHCEN))
      PFRMZ  = PFRMXY*SIGMZ/SIGMXY
C
C     X-DIRECTION:
      PUFRGX = PFRMXY/DELX
C     Y-DIRECTION:
      PUFRGY = PFRMXY/DELY
C     Z-DIRECTION:
      PUFRGZ = PFRMZ/DELZ
C
C     DETERMINING START AND STOP GRID POINTS FOR ACCUMULATION OF
C     THE PUFFS IN QUESTION:
C
      ISTRTX = PTBL3 - PUFRGX + 1
      ISTOPX = PTBL3 + PUFRGX
      ISTRTY = PTBL4 - PUFRGY + 1
      ISTOPY = PTBL4 + PUFRGY
      ISTRTZ = PTBL5 - PUFRGZ + 1
      ISTOPZ = PTBL5 + PUFRGZ
C
C     CONTROL FOR EXCEEDING GRID DIMENSIONS
C
      IF(ISTRTX.LT.XSB) ISTRTX=XSB
      IF(ISTOPX.GT.XLB) ISTOPX=XLB
      IF(ISTRTY.LT.YSB) ISTRTY=YSB
      IF(ISTOPY.GT.YLB) ISTOPY=YLB
      IF(ISTRTZ.LT.ZSB) ISTRTZ=ZSB
      IF(ISTOPZ.GT.ZLB) ISTOPZ=ZLB
C
```

```
C     UPPER LIMIT IN CASE OF SPECIFIED MIXING DEPTH:ZM
      IF(.NOT.NOMXDP   .AND. ISTOPZ .GT. ZMG ) ISTOPZ = ZMG
      IF(ISTRTZ .GT.ISTOPZ) GO TO 1500
C
C     CALCULATE CONTRIBUTIONS TO SURROUNDING GRIDPOINTS
C
C     PRELIMINAR CALCULATIONS:
C     SIGMAS IN GRIDUNITS:
      SIGGX = SIGMXY/DELX

      SIGGY = SIGMXY/DELY
      SIGGZ = SIGMZ /DELZ
C
C     CALCULATING DENOMINATOR UNDER EXP-SIGN:
      SIGGX2 = (SIGGX**2)*(-2)
      SIGGY2 = (SIGGY**2)*(-2)
      SIGGZ2 = (SIGGZ**2)*(-2)
C
C     LOOPING THRU ALL GRIDPOINTS OF INTEREST:
C
      DO 1500  KG = ISTRTZ,ISTOPZ
      ZG2NEG = (KG-PTBL5)**2
      PCHI1  = PCHCEN * EXP(ZG2NEG/SIGGZ2)
      IF(GRRFLX) PCHI1 = PCHI1 + PCHCEN*REFLEC*EXP((KG+PTBL5)**2/SIGGZ2)
      IF(NOMXDP) GO TO 1295
      IF((PTBL5+PUFRGZ) .LT. ZMG) GO TO 1295
      ZG2MX  = (KG+PTBL5-2*ZMG)**2
      PCHI1  = PCHI1 + PCHCEN*EXP(ZG2MX/SIGGZ2)
 1295 DO 1500  IG = ISTRTX,ISTOPX
      XG2 = (IG-PTBL3)**2
      DO 1500  JG = ISTRTY,ISTOPY
      YG2 = (JG-PTBL4)**2
C
C     INDIVIDUAL PUFFS CONTRIBUTION : PCHI,GRAM/M**3
C
      PCHI   = PCHI1 * EXP(XG2/SIGGX2 + YG2/SIGGY2)
C
      IF(PCHI .LT. CHEMIN) GO TO 1500
C     ACCUMULATING IN GRIDPOINTS:
      CHI(IG+1,JG+1,KG+1) = CHI(IG+1,JG+1,KG+1) + PCHI
C
 1500 CONTINUE
C
C     END OF CONCENTRATION CALCULATIONS
C
C
C     ADVANCE IN PUFF TABLE (J) DURING BASIC ADVECTION STEP
      J = J + 1
```

```
      IF(PTABEL(I2,J,1) .NE. 0) GO TO 1260
 1300 CONTINUE
C     END OF ADVECTION SECTION
C
C
C
C******************************************************************
C**********************  OUTPRINT SECTION  ************************
C******************************************************************
C
C     SECTION FOR OUTPRINTING PUFF POSITIONS IN GRID.
C
C
C     SKIPPING PUFF POSITION PLOT IF SPECIFIED:
C
C     TOTTIM TRUNCATED TO INTEGER NUMBER: ITOTIM.
      ITOTIM = TOTTIM
      IF(MOD(ITOTIM,MAPTIM) .NE. 0) GO TO 1600
 1301 FORMAT(1H1)
 1310 FORMAT(1H0,30X,'PLOT OF CURRENT PUFF POSITIONS',I6,
     &'SEC AFTER START OF RELEASE.')
 1320 FORMAT(1H0,30X,'TOTALLY',I6,'PUFFS HAVE BEEN RELEASED AND',I6,
     &'PUFFS HAVE LEFT THE GRID.')
 1325 FORMAT(1H+,24X,105A1)
 1326 FORMAT(1H ,17X,1HI,105A1)
 1327 FORMAT(1H ,I11,5X,2H +,105A1)
 1328 FORMAT(1H+,25X,105A1)
C
C
C     SUMMING UP ALL PUFFS RELEASED: SUMPUF
      SUMPUF = 0
      DO 1329 NPUF = 1,NRMULT
 1329 SUMPUF = SUMPUF + TPUFFS(NPUF)
C
      WRITE(6,1301)
      WRITE(6,1310) ITOTIM
      WRITE(6,1)
      WRITE(6,1320) SUMPUF,LEAVE
      WRITE(6,1)
      WRITE(6,1)
C
      IF(ICOLS .GT. 10) GO TO 1595
      WRITE(6,870) (I2,I2  =  XSB,XLB)
      WRITE(6,1)
      WRITE(6,912) HORFRM
      WRITE(6,913) VERFRM,VERFRM
C
C     LOOPING THRU ALL Y-VALUES OF GRID, STARTING UPPERMOST:
```

```
      MAX = JROWS - 1                                                   PUF07960
C     OUTER LOOP THRU INTEGER Y-VALUES:
      NY8=MAX+1
      DO 1350   NY9=1,NY8
      I2=NY9-1
      MAXMI2 = MAX - I2                                                 PUF08000
      WRITE(6,1327) MAXMI2 , VERPLS                                     PUF08010
C                                                                       PUF08020
C     PLOTTING SOURCE POSITIONS                                         PUF08030
      K1 = 0                                                            PUF08040
      DO 1330  J = 1,NRMULT                                             PUF08050
      IF(MAX-I2 .NE. YSOURC(J) )  GO TO 1330                            PUF08060
C     NUMBER OF SOURCES IN MAINLINE: K1                                 PUF08070
      K1 = K1 + 1                                                       PUF08080
      CALL ISPACE(XSOURC(J),J)                                          PUF08090
C     SOURCE POSITIONS IN EACH MAINLINE: XINT(K1)
      XINT(K1) = 10*XSOURC(J)                                           PUF08110
 1330 CONTINUE                                                          PUF08120
C                                                                       PUF08130
C     LOOPING 9 LINES DOWN TO NEXT MAINLINE:                            PUF08140
      DO 1345 NY10=1,10
      IDECI=NY10-1
      YLINE = 10*(MAX - I2) - IDECI + 10                                PUF08160
      IF(IDECI .GE. 1) WRITE(6,1326) VERFRM                             PUF08170
C                                                                       PUF08180
C     SCANNING THRU WHOLE PUFF TABLE                                    PUF08190
      DO 1340 II = 1,NRMULT                                             PUF08200
      J = 0                                                             PUF08210
 1335 J = J + 1                                                         PUF08220
      IF(PTABEL(II,J,1) .EQ. 0) GO TO 1340                              PUF08230
C     TRUNCATING Y-VALUE OF PUFF TO INTEGER:                            PUF08240
      YINT = PTABEL(II,J,4)*10 + 10.5                                   PUF08250
C                                                                       PUF08260
C     PRINTING "*" IN GRIDFRAME IF X-POSITION OF PUFF NOT COINCIDE      PUF08270
C     WITH ONE OF THE SOURCE POSITIONS                                  PUF08280
      IF( (YINT .NE. YLINE) .OR. (IDECI .NE. 0) ) GO TO 1338            PUF08290
      COINCD = .FALSE.                                                  PUF08300
C     INTEGER VALUE OF PUFFS X-POSITION: XINTPF                         PUF08310
      XINTPF = PTABEL(II,J,3) * 10 + .5                                 PUF08320
      DO 1336 KK = 1,K1                                                 PUF08330
 1336 IF(XINTPF .EQ. XINT(KK)) COINCD = .TRUE.                          PUF08340
      IF(COINCD) GO TO 1335                                             PUF08350
      STRING(XINTPF + 1) = SN1                                          PUF08360
      GO TO 1335                                                        PUF08370
C                                                                       PUF08380
 1338 IF(YINT .NE. YLINE) GO TO 1335                                    PUF08400
C     PRINTING PUFF POSITIONS BETWEEN Y-GRID LINES:
      XINTPF = PTABEL(II,J,3) * 10 + .5                                 PUF08420
```

```
      STRING(XINTPF + 1) = SN1                                          PUF08430
      GO TO 1335                                                        PUF08440
C                                                                       PUF08450
 1340 CONTINUE                                                          PUF08470
C     END OF PUFF TABLE LOOP.                                           PUF08480
C                                                                       PUF08490
C                                                                       PUF08500
      WRITE(6,1325) STRING                                              PUF08510
      DO 1342 NST = 1,105                                               PUF08520
 1342 STRING(NST)    = BL                                               PUF08530
C                                                                       PUF08540
 1345 CONTINUE                                                          PUF08550
C                                                                       PUF08570
C     RESET "SOURCE IN LINE COUNTER" XINT(KK)                           PUF08580
      DO 1349 KK=1,10                                                   PUF08590
 1349 XINT(KK) = -1                                                     PUF08600
C                                                                       PUF08610
 1350 CONTINUE                                                          PUF08620
C     END OF PUFF POSITION PLOT.                                        PUF08630
      WRITE(6,1)                                                        PUF08640
      WRITE(6,912) HORFRM                                               PUF08650
 1400 CONTINUE                                                          PUF08660
C                                                                       PUF08670
C     PLOTTING PUFFS IN "Y-Z FRAME";FOR COMMENTS REFER TO THE EQUI-     PUF08690
C     VALENT "Y-X FRAME" PLOTTING DESCRIBED ABOVE.                      PUF08700
C                                                                       PUF08710
      WRITE(6,1)                                                        PUF08720
      WRITE(6,1)                                                        PUF08730
      WRITE(6,1)                                                        PUF08740
  881 FORMAT(1H+,20X,I5,10H SOURCE(S))                                  PUF08750
      WRITE(6,871) (I2,I2=ZSB,ZLB)                                      PUF08760
      WRITE(6,1)                                                        PUF08770
C     HRFRMZ : STRING CONTAINING HORIZONTAL GRID FRAME                  PUF08780
      DO 1410 N = 1,105                                                 PUF08790
      VRFRMZ(N) = BL                                                    PUF08800
      VRPLSZ(N) = BL                                                    PUF08810
      PARENT(N) = BL                                                    PUF08820
 1410 HRFRMZ(N) = BL                                                    PUF08830
      NY11=MFZ*10
      DO 1418 IHFZ = MSZ,NY11,10                                        PUF08840
      NY12=IHFZ+4
      DO 1411 MN = IHFZ,NY12
 1411 HRFRMZ(MN) = SN4                                                  PUF08860
      HRFRMZ(IHFZ+5) = SN2                                              PUF08870
      NY13=IHFZ+6
      NY14=IHFZ+9
      DO 1416 MM=NY13,NY14
 1416 HRFRMZ(MM) = SN4                                                  PUF08890
```

```
 1418 CONTINUE
      WRITE(6,912) HRFRMZ
      PARENT(10*ZMG + 1) = SN6
      VRFRMZ(10*MFZ + 3) = SN5
      VRPLSZ(10*MFZ + 3) = SN3
      WRITE(6,1326) VRFRMZ
      WRITE(6,1326) VRFRMZ
      MAX = JROWS-1
      DO 1445 NY15=1,JROWS
      I2=NY15-1
      MAXMI2 = MAX - I2
      WRITE(6,1327) MAXMI2,VRPLSZ
      K1=0
      DO 1430 J=1,NRMULT
      IF(MAX-I2 .NE. YSOURC(J)) GO TO 1430
      K1 = K1+ 1
 1430 CONTINUE
C     WRITING NUMBER OF SOURCES IN EACH Y-GRIDLINE: VN SOURCE(S)
      IF(K1 .GT. 0) WRITE(6,881) K1
      DO 1445 NY16=1,10
      IDECI=NY16-1
      YLINE = 10*(MAX-I2) - IDECI +10
      IF(IDECI.GE.1) WRITE(6,1326) VRFRMZ
C     ILLUSTRATING MIXING DEPTH IN Y-Z FRAME:
      IF(ZMG.GT.0) WRITE(6,1328) PARENT
      DO 1440 II = 1,NRMULT
      J=0
 1435 J = J + 1
      IF(PTABEL(II,J,1) .EQ. 0) GO TO 1440
      YINT = PTABEL(II,J,4) *10 + 10.5
      IF(YINT .NE. YLINE) GO TO 1435
C
      ZINTPF = PTABEL(II,J,5)* 10 + .5
      STRING(ZINTPF + 1) = SN1
      GO TO 1435
C
 1440 CONTINUE
C
      WRITE(6,1325) STRING
      DO 1442 NST = 1,105
 1442 STRING(NST) = BL
C
 1445 CONTINUE
C
      WRITE(6,1)
      WRITE(6,912) HRFRMZ
C
C     SECTION FOR OUTPRINTING GRID CONCENTRATIONS
```

```
C                                                                       PUF09360
C     SKIPPING CONCENTRATION PRINTING IF SPECIFIED IN PRIMDA.
      IF(ABC(9).EQ. NO ) GO TO 1600                                     PUF09380
C                                                                       PUF09390
 1510 FORMAT(1H0,49X,37H PRINT OF CURRENT GRID CONCENTRATIONS,/50X      PUF09410
     1,I6,29H SEC. AFTER START OF RELEASE.)                             PUF09420
 1520 FORMAT(1H0,49X,32H GRIDCONCENTRATION IN THE PLANE:,/51X,3HZ =F6.2 PUF09430
     1,25H METER ABOVE THE SURFACE.)                                    PUF09440
 1525 FORMAT(I11,8X,10E10.2)                                            PUF09450
C                                                                       PUF09460
      WRITE(6,1301)                                                     PUF09470
      WRITE(6,1510) ITOTIM                                              PUF09480
      WRITE(6,1)                                                        PUF09490
      WRITE(6,1)                                                        PUF09500
C                                                                       PUF09510
C     LOOP THRU ALL Z LEVELS                                            PUF09520
C                                                                       PUF09530
      DO 1550  KC=1,KPLANS                                              PUF09540
      DEMKM1 = DELZ*(KC-1)                                              PUF09550
      WRITE(6,1520) DEMKM1                                              PUF09560
      WRITE(6,1)                                                        PUF09570
      WRITE(6,870)  (IC,IC = XSB,XLB)                                   PUF09580
C     PRINTING EACH LINE IN CONCENTRATION TABLE:                        PUF09590
      DO 1560 JC = 1,JROWS                                              PUF09600
      JJC     = JROWS - JC                                              PUF09610
      JC1     = JJC + 1                                                 PUF09620
      WRITE(6,1525) JJC,(CHI(IC,JC1,KC) , IC = MSX,MFX)                 PUF09630
      DO 1551 IC=MSX,MFX                                                PUF09640
 1551 CPLOT(IC,JC1)=CHI(IC,JC1,KC)                                      PUF09650
 1552 FORMAT(5X,10E10.2)                                                PUF09670
 1560 CONTINUE                                                          PUF09680
C
C     KC IS THE NO. OF LEVELS PRINTED...HERE CONTROLS WHICH
C     LEVELS ARE CONTOURED.
      IF (KC.EQ.1) CALL DRAW(CPLOT,10,17)
 1550 CONTINUE                                                          PUF09690
      WRITE(6,1)                                                        PUF09710
      WRITE(6,1)                                                        PUF09720
C                                                                       PUF09730
      GO TO 1600                                                        PUF09750
C                                                                       PUF09760
 1590 FORMAT(95H PUFF POSITION PLOT AND GRID CONCENTRATION PRINTING AR  PUF09770
     1E SUPPRESSED BECAUSE "ICOLS" EXCEED 10.)                          PUF09780
 1595 WRITE(6,1590)                                                     PUF09790
C                                                                       PUF09800
 1600 CONTINUE                                                          PUF09830
C     END OF GRID CONCENTRATION PRINTING SECTION                        PUF09840
C                                                                       PUF09870
```

```
C      END OF OUTPRINT SECTION.
C
 5000 CONTINUE
C      END OF ADVECTION STEPS DURING CURRENT WIND FIELD SPECIFICATION
C
C      END OF CALCULATION PART.
       I = I+2
       IF(I.GE.14) GO TO 1135
C      RETURN FETCHING NEW ANGLE,SPEED:
       GO TO 1150
C
C      OUTPUT DIAGNOSTICS :
 1000 FORMAT(37H X AND/OR Y COORDINATES OF SOURCE NR: ,I5,12H IS OFF GRID
     1)
 1005 FORMAT(72H FORMAT ERROR IN SECOND WINDDA CARD:MISSING OR WRONG PL
     1ACED #-CARACTER|)
 1010 FORMAT(55H BAD SPECIFICATION OF PUFF RELEASE AND ADVECTION STEP,
     15H TAU=,I5,6X,7H NTADV=,I5)
 1015 FORMAT(60H BAD SPECIFICATION OF WINDAVERAGING TIME AND ADVECTION S
     1TEP:,8H WINDAV=,I4,7X,7H NTADV=,I5)
 1025 FORMAT(64H ERROR IN WINDDATA-SPECIFICATION OF:  *SPEED,"ANGLE, AFT
     1ER THE:,I6,24H.STABILITY SPECIFICATION)
 1030 FORMAT(86H MISSING STABILITY CLASS SPECIFICATION,THE LAST SPECIFI
     1ED STABILITY CLASS NUMBER WAS,I6)
 1035 FORMAT(72H FORMAT ERROR IN SECOND PUFFDA-CARD: MISSING OR WRONG PL
     1ACED #-CARACTER|)
 1040 FORMAT(16H MISMATCH IN THE,I5,22H. SOURCE SPECIFICATION)
 1045 FORMAT(12H SOURCE NR:,I5,72H HAS MORE THN 100 CONTRIBUTING PUFFS
     1IN THE GRID,TAU MUST BE INCREASED.)
 1050 FORMAT(72H FORMAT IN SECOND INTSDA-CARD; MISSING OR WRONG PL
     1ACED #-CARACTER|)
 1055 FORMAT(41H ZM MUST BE AN INTEGER MULTIPLUM OF DELZ|)
 1060 FORMAT(47H MIXING LAYER DEPTH EXCEEDS Z DIMENSION OF GRID)
C
C
C
 8870 WRITE(6,1060)
       GO TO 9999
 8880 WRITE(6,1055)
       GO TO 9999
 8890 WRITE(6,1050)
       GO TO 9999
 8900 WRITE(6,1045) I
       GO TO 9999
 8910 WRITE(6,1040) I
       GO TO 9999
 8920 WRITE(6,1035)
       GO TO 9999
```

```
 8930 WRITE(6,1140)
      GO TO 9999
 8940 NRM1    = NRSTAB - 1
      WRITE(6,1030) NRM1
      GO TO 9999
 8950 WRITE(6,1025) NRSTAB
      GO TO 9999
 8970 WRITE(6,1015) WINDAV,NTADV
      GO TO 9999
 8980 WRITE(6,1010) TAU,NTADV
      GO TO 9999
 8990 WRITE(6,1005)
      GO TO 9999
 9000 WRITE(6,1000) I
C
 9999 CONTINUE
      CALL EFRAME
      STOP
      END
C
C
C
C
C
C
      SUBROUTINE SIGRIS(HGN,SIGXY,SIGZ)
C
C     THE SUBROUTINE "SIGRIS" (SIGMA-RISE) CALCULATES THE INCREMENT
C     IN SIGMA-XY AND SIGMA-Z DURING EACH BASIC ADVECTION STEP.
C     FURTHER, THE SUBROUTINE ESTIMATES PLUMERISE ASSOCIATED WITH
C     BOUYANCY IN THE EFFLUXES.
C
C     FOR Z-COORDINATES OF PUFFS: HEIGHT , GRID UNITS(N) : HGN
C
      COMMON HEATFX(25),I2,DMS,POINT,INTENS(14),STABPA,FBUFLX
     1,UNN, CONST1,DELZ
      INTEGER POINT
      REAL INTENS
C     CALCULATING GROWTH RATES FOR SIGMAS: DSIGDS
C     DEFINING EXPERIMENTAL FITTING CONSTANT: FITCST
      FITCST = 2.0
      DSIGDS = .22 * INTENS(POINT)
      DSIGDS = DSIGDS * FITCST
      SIGXY  = SIGXY + DSIGDS * DMS
      SIGZ   = SIGZ  + DSIGDS * DMS
C
C     CALCULATING PLUME-RISE INCREMENT:
C
```

```
C      FI : BUDYANT FLUX FROM SOURCE I                                  PUF10930
C      AFTER BRIGGS:                                                    PUF10940
C      F = 8.9 < M**4/SEC**3> * Q < MWATT >                             PUF10950
C      HEATFX(I2) UNITS ARE KW                                          PUF10960
       FI = 8.9 * 0.001 * HEATFX(I2)                                    PUF10970
C                                                                       PUF10980
       IF(STABPA.LE.0.0) GO TO 2501                                     PUF11010
C      MAXIMUM PLUME LIFT IN STABLE ATMOSPHERE:HSMAX.                   PUF11020
       HSMAXG = 2.9*(FI/(UNN*STABPA))**(1./3) / DELZ                    PUF11030
 2501  CONTINUE                                                         PUF11040
C                                                                       PUF11050
C      CALCULATING PLUME HEIGHT AFTER FILFILLED ADVECTION STEP          PUF11060
C      HGRID(N+1) : HGNP1.                                              PUF11070
       HGNP1 = HGN + CONST1*SQRT(FI/HGN)/((DELZ *UNN)**1.5)*DMS         PUF11080
       IF(STABPA.LE.0.0) GO TO 2510                                     PUF11090
       IF(HGNP1.GT.HSMAXG) GO TO 2520                                   PUF11100
 2510  HGN = HGNP1                                                      PUF11110
 2520  CONTINUE                                                         PUF11120
       RETURN                                                           PUF11130
       END                                                              PUF11140
C                                                                       PUF11150
C                                                                       PUF11160
C                                                                       PUF11170
C                                                                       PUF11180
       SUBROUTINE ISPACE(ITENFT,INR)                                    PUF11190
C                                                                       PUF11200
C      THE SUBROUTINE "ISPACE" MAKES VARIABLE TABULATING POSSIBLE       PUF11210
C      IN CONNECTION WITH FRAMEPLOTS.
C                                                                       PUF11230
C      ITENFT: NUMBER OF TEN SPACES,THE FIGURE IN QUESTION HAS TO       PUF11240
C              BE MOVED RIGHTMOST.                                      PUF11250
C                                                                       PUF11260
C      INR   : INTEGER NUMBER TO BE PRINTED.                            PUF11270
C                                                                       PUF11280
   10  FORMAT(1H+,19X,I6)                                               PUF11290
   20  FORMAT(1H+,29X,I6)                                               PUF11300
   30  FORMAT(1H+,39X,I6)                                               PUF11310
   40  FORMAT(1H+,49X,I6)                                               PUF11320
   50  FORMAT(1H+,59X,I6)                                               PUF11330
   60  FORMAT(1H+,69X,I6)                                               PUF11340
   70  FORMAT(1H+,79X,I6)                                               PUF11350
   80  FORMAT(1H+,89X,I6)                                               PUF11360
   90  FORMAT(1H+,99X,I6)                                               PUF11370
  100  FORMAT(1H+,109X,I6)                                              PUF11380
C                                                                       PUF11390
C                                                                       PUF11400
       IF(ITENFT.NE.0) GO TO 1                                          PUF11410
       WRITE(6,10) INR                                                  PUF11420
```

```
      RETURN
    1 IF(ITENFT.NE.1) GO TO 2
      WRITE(6,20) INR
      RETURN
    2 IF(ITENFT.NE.2) GO TO 3
      WRITE(6,30) INR
      RETURN
    3 IF(ITENFT.NE.3) GO TO  4
      WRITE(6,40) INR
      RETURN
    4 IF(ITENFT.NE.4) GO TO  5
      WRITE(6,50) INR
      RETURN
    5 IF(ITENFT.NE.5) GO TO  6
      WRITE (6,60) INR
      RETURN
    6 IF(ITENFT.NE.6) GO TO  7
      WRITE(6,70) INR
      RETURN
    7 IF(ITENFT.NE.7)GO TO    8
      WRITE(6,80) INR
      RETURN
    8 IF(ITENFT.NE.8) GO TO  9
      WRITE(6,90) INR
      RETURN
    9 IF(ITENFT.NE.9) GO TO 1000
      WRITE(6,100) INR
 1000 RETURN
      END
C
C
C
      SUBROUTINE DRAW (A,M,N)
C
C    THIS SUBROUTINE CONVERTS THE GRID CONCENTRATIONS TO LOG VALUES
C    THEN SMOOTHES AND CONTOURES THE PUFF ARRAY...
C
      DIMENSION A(M,N)
C
C     A IS THE ARRAY TO BE SMOOTHED AND CONTOURED
C     M,N IS THE DIMENSION OF ARRAY A
C
      DO 1553 JC=1,N
      JJC=N-JC
      JC1=JJC+1
      DO 1554 IC=1,M
      IF(A(IC,JC1).LT. .1E-12) A(IC,JC1)=0.0
      A(IC,JC1)=A(IC,JC1)*1.E13
```

```
      IF(A(IC,JC1).EQ.0.0) GO TO 1554
      A(IC,JC1)=ALOG10(A(IC,JC1))
 1554 CONTINUE
 1553 WRITE(6,1555) (A(IC,JC1),IC=1,M)
 1555 FORMAT(5X,10E10.2)
C     SMOOTH ARRAY
      MM1=M-1
      NM1=N-1
      DO 200 J=1,N
      TEMP=A(1,J)
      DO 100 I=2,MM1
      TEMP1=A(I,J)
      A(I,J)=.20*(TEMP+3.*TEMP1+A(I+1,J))
      TEMP=TEMP1
  100 CONTINUE
  200 CONTINUE
      DO 400 I=1,M
      TEMP=A(I,1)
      DO 300 J=2,NM1
      TEMP1=A(I,J)
      A(I,J)=.20*(TEMP+3.*TEMP1+A(I,J+1))
      TEMP=TEMP1
  300 CONTINUE
  400 CONTINUE
      DO 1559 J=1,17
 1559 WRITE (6,1555) (A(I,J),I=1,10)
      CALL SET (.1,.31,.2,.58,0.,1.,0.,1.,1)
      CALL CONREC(A,10,10,17,0.,0.,1.,-1,-1,0)
      CALL TICK4(5,8,5,8)
      CALL PERIM(9,0,16,0)
      CALL FRAME
      RETURN
      END
C
C
C
C
      SUBROUTINE RSPACE(ITENFT,RNR)
C
C     THIS SUBROUTINE HAS THE SAME PURPOSE FOR REAL FIGURES, AS
C     ISPACE HAS FOR INTEGER FIGURES.
C
C     RNR    : REAL NUMBER TO BE PRINTED.
C
   10 FORMAT(1H+,19X,F6.1)
   20 FORMAT(1H+,29X,F6.1)
   30 FORMAT(1H+,39X,F6.1)
   40 FORMAT(1H+,49X,F6.1)
```

```
   50 FORMAT(1H+,59X,F6.1)
   60 FORMAT(1H+,69X,F6.1)
   70 FORMAT(1H+,79X,F6.1)
   80 FORMAT(1H+,89X,F6.1)
   90 FORMAT(1H+,99X,F6.1)
  100 FORMAT(1H+,109X,F6.1)
C
C
      IF(ITENFT.NE.0) GO TO 1
      WRITE(6,10) RNR
      RETURN
    1 IF(ITENFT.NE.1) GO TO 2
      WRITE(6,20) RNR
      RETURN
    2 IF(ITENFT.NE.2) GO TO 3
      WRITE(6,30) RNR
      RETURN
    3 IF(ITENFT.NE.3) GO TO 4
      WRITE(6,40) RNR
      RETURN
    4 IF(ITENFT.NE.4) GO TO 5
      WRITE(6,50) RNR
      RETURN
    5 IF(ITENFT.NE.5) GO TO 6
      WRITE(6,60) RNR
      RETURN
    6 IF(ITENFT.NE.6) GO TO 7
      WRITE(6,70) RNR
      RETURN
    7 IF(ITENFT.NE.7) GO TO 8
      WRITE(6,80) RNR
      RETURN
    8 IF(ITENFT.NE.8) GO TO 9
      WRITE(6,90) RNR
      RETURN
    9 IF(ITENFT.NE.9) GO TO 1000
      WRITE(6,10) RNR
 1000 RETURN
      END
```

Risø National Laboratory **Risø - M - 2361**

Risø - M -

Title and author(s)

Description of the Risø puff diffusion model

Torben Mikkelsen

Date October 1982

Department or group

Physics Dept.

Group's own registration number(s)

43 pages + tables + illustrations

Abstract

The Risø National Laboratory, Roskilde, Denmark, atmospheric puff dispersion model is described. This three-dimensional model simulates the release of Gaussian pollutant puffs and predicts their concentration as they are diffused and advected downwind by a horizontally homogeneous, time-dependent wind. Atmospheric characteristics such as turbulence intensity, potential temperature gradient, buoyant heat flux and maximum mixing depth have been considered.

Copies to

Available on request from Risø Library, Risø National Laboratory (Risø Bibliotek), Forsøgsanlæg Risø), DK-4000 Roskilde, Denmark
Telephone: (02) 37 12 12, ext. 2262. Telex: 43116